A Fast Geometric Decoding Algorithm for MIMO Systems

Z. Y. Shao, S. W. Cheung and T. I. Yuk
Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong
zyshao@eee.hku.hk, swcheung@eee.hku.hk and tiyuk@eee.hku.hk

Abstract — **One of the practical challenges of Wireless communications over Multiple-input Multiple-output (MIMO) channels is to reduce the complexity of the decoding algorithm without sacrifice in performance degradation. Geometric decoding (GD) is an emerging technique which can offer substantially lower complexity while maintaining about the same error rate as that of Maximum likelihood (ML) decoding. In this paper we propose a fast GD algorithm which uses a hyper ellipsoid to reduce the searching region. Simulation results on different MIMO systems transmitting 64QAM show that the proposed GD algorithm can achieve about the same error-rate performance as an ML decoder, yet having less than 5% of the decoding complexity of an ML decoder.**

Keywords — **MIMO, Geometric decoding, low complexity.**

1. Introduction

Wireless communications over Multiple-input Multiple-output (MIMO) channels provides an efficient way to increase the spectral efficiency by exploiting the space [1, 2]. To achieve this advantage, an extremely complicated Maximum-likelihood (ML) decoder which employs exhaustive searching is needed to be used at the receiver. However, when the numbers of transmit and receive antennas are large or the modulation levels are high, the ML decoder will be too complicated to implement in practice. Therefore, researchers have been working hard and proposing different techniques to reduce the complexity of decoding methods for the MIMO channels. Geometric decoding (GD) is one of these proposed methods [3, 4].

GD is a search-based algorithm. By using the channel matrix, GD constructs an ellipsoid paraboloid on which all the transmit symbol vectors are lying. In [5], the authors proposed a GD algorithm for QAM and PAM. The complexity is very low compared to that of an ML algorithm. However, when the numbers of transmit and receive antennas increase, the block-error-rate (BLER) performances degrade in comparison with those of using an ML decoder.

In this paper, we propose a simple approach to implement a fast GD algorithm for use in the MIMO systems over fading channels. The proposed GD algorithm first constructs an elliptic paraboloid using the channel matrix which is assumed to be known at the receiver. A group of hyper ellipsoids can be projected from the elliptic paraboloid. All possible transmit symbol vectors are lying on these hyper ellipsoids. To reduce the number of symbol vectors used for searching, we propose to restrict the search to the symbol vectors only inside the hyper ellipsoid which is projected from the elliptic paraboloid containing the zero forcing solution. It is very difficult to identify the symbol vectors inside this hyper ellipsoid. To solve this problem, we first set up a hyper rectangle which tightly encloses the hyper ellipsoid. Since this hyper rectangle does not have the axes in parallel with those of the lattice, it is not easy to use. So to simplify the job, we propose to set up another hyper rectangle having the axes in parallel with those of the lattice to enclose this hyper rectangle. Complexity is measured by the number of symbol vectors used in the decoding process to obtain the final solution. Monte Carlo simulation results show the complexity of our proposed GD algorithm is substantially lower than that of an ML decoding algorithm. It is an optimum GD algorithm in the sense that it minimizes the BLERs identical to what an ML decoding algorithm does.

The remainder of this paper is organized as follows. Section 2 describes the model of MIMO system used for studies. The general description on the concept of GD and our proposed GD implementation are presented in section 3. Simulation results and discussions are given in section 4. Finally, conclusions are drawn in section 5.

2. MIMO Channel

Consider an uncoded MIMO system with N transmit and N receive antennas over a fading channel. The received signal matrix is given by

$$\mathbf{r} = \mathbf{H}\mathbf{x} + \mathbf{w} \tag{1}$$

where $\mathbf{r} = [r_1, r_2, ..., r_N]^T$ is the N-dimensional received signal vector, $\mathbf{x} = [x_1, x_2, ..., x_N]^T$ a symbol vector in the transmit lattice, with $[]^T$ denoting vector or matrix transposition, $\mathbf{H}$ is the channel matrix, with elements h_{ij} representing the transfer function from the j-th transmit antenna to i-th receive antenna, and $\mathbf{w} = [w_1, w_2, ..., w_N]^T$ is an independently and identically distributed (i.i.d.) zero-mean Gaussian noise vector with elements having a fixed variance. The elements in (1) could be either complex values or real values, e.g. for PAM and QAM transmission, the elements in (1) are complex and real values, respectively. However, to implement the detection process, it is usually more efficient to convert the complex system into a real one and process the real values. So even for QAM transmissions, the elements in the matrices of (1) will also be transformed into a real matrix equation:

$$\begin{bmatrix} \mathbf{Re}(\mathbf{r}) \\ \mathbf{Im}(\mathbf{r}) \end{bmatrix} = \begin{bmatrix} \mathbf{Re}(\mathbf{H}) & \mathbf{-Im}(\mathbf{H}) \\ \mathbf{Im}(\mathbf{H}) & \mathbf{Re}(\mathbf{H}) \end{bmatrix} \begin{bmatrix} \mathbf{Re}(\mathbf{x}) \\ \mathbf{Im}(\mathbf{x}) \end{bmatrix} + \begin{bmatrix} \mathbf{Re}(\mathbf{w}) \\ \mathbf{Im}(\mathbf{w}) \end{bmatrix} \quad (2)$$

with Re(·) and Im(·) being the real and imaginary parts, respectively, of $(\cdot)$. As it is much easier to explain the GD process in the real dimensional space, therefore, without lost of generality, real value vectors and matrices are assumed in (1) for discussions in the rest of the paper.

Assuming the channel matrix **H** in (1) is known at the receiver, for minimizing the BLER of the system, an ML decoder will select the symbol vector in the lattice with the minimum Euclidean distance. The transmitted symbol vector is detected as

$$\mathbf{x_{ML}} = \arg\min_{\mathbf{x}\in\Omega} \left\|\mathbf{r} - \mathbf{Hx}\right\|^2 \quad (3)$$

where $\mathbf{r}, \mathbf{x} \in \Re^N$, $\mathbf{H} \in \Re^{N\times N}$ and Ω is the set of all possible symbols in the transmitted lattice.

3. Geometric Decoding (GD)

The Euclidean distance $\left\|\mathbf{r} - \mathbf{Hx}\right\|^2$ in (3) can be written as [5]:

$$\begin{aligned} f(\mathbf{x}) &= \left\|\mathbf{r} - \mathbf{Hx}\right\|^2 \quad (4) \\ &= (\mathbf{x} - \mathbf{x_c})^{\mathbf{T}} \mathbf{M}^{-1} (\mathbf{x} - \mathbf{x_c}) \\ &= (\mathbf{x} - \mathbf{x_c})^{\mathbf{T}} \left(\mathbf{H}^{\mathbf{T}}\mathbf{H}\right)^{-1} (\mathbf{x} - \mathbf{x_c}) \end{aligned}$$

which is an elliptic paraboloid in an *N+1* dimensional space with the axis perpendicular to an *N* dimensional subspace spanned by the symbol vectors in Ω. In (4), $\mathbf{x_c}$ is given by

$$\mathbf{x_c} = \mathbf{H}^{-1}\mathbf{r} \quad (5)$$

which is the global minimum point of the elliptic paraboloid and is located on the subspace spanned by the symbol vectors in Ω as shown in Fig. 1. It can be seen from (4) that the function $f(\mathbf{x})$ reaches its minimum value at $\mathbf{x_c}$, i.e.,

$$f(\mathbf{x})_{\min} = f(\mathbf{x_c}) = 0 \quad (6)$$

Using eigenvalue decomposition, the matrix **M** in (4) can be written as:

$$\mathbf{M} = \left(\mathbf{H}^{\mathbf{T}}\mathbf{H}\right)^{-1} = \left(\mathbf{V}\Lambda\mathbf{V}^{T}\right)^{\mathbf{T}} \quad (7)$$

where $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_N)$ are the eigenvalues of **M** arranged in descending order, and **V** is the corresponding eigenvector denoted by

$$\begin{aligned} \mathbf{V} &= [\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}, \cdots, \mathbf{v_N}] \quad (8) \\ &= \begin{bmatrix} v_{11} & v_{21} & v_{31} & \cdots & v_{N1} \\ v_{12} & v_{22} & v_{32} & \cdots & v_{N2} \\ v_{13} & v_{23} & v_{33} & \cdots & v_{N3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v_{1n} & v_{2n} & v_{3n} & \cdots & v_{NN} \end{bmatrix} \end{aligned}$$

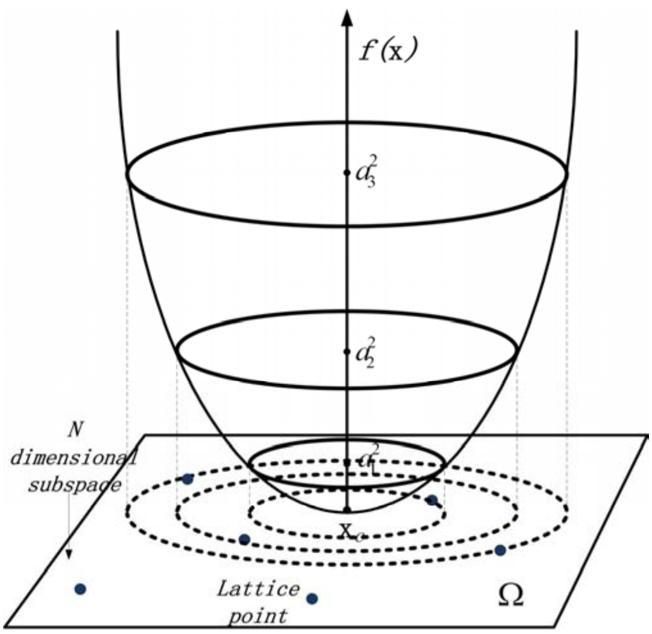

Fig. 1 Elliptic paraboloid in an *N*+1 dimensional space with axis perpendicular to a subspace spanned by the vectors in Ω

From Fig. 1, it can be seen that the horizontal cross section of the elliptic paraboloid (4) is a hyper ellipsoid given by

$$f(\mathbf{x}) = a^2 \quad (9)$$

which has its center at a distance a^2, where $a \in \Re$, from the global minimum point $\mathbf{x_c}$. Thus, (9) can be rewritten as:

$$\frac{(x_1 - x_{c1})^2}{a^2\lambda_1} + \frac{(x_2 - x_{c2})^2}{a^2\lambda_2} + \ldots + \frac{(x_N - x_{cN})^2}{a^2\lambda_N} = 1 \quad (10)$$

where x_i and x_{ci} are the *i*-th components of vectors **x** and $\mathbf{x_c}$, respectively, and λ_i is the *i*-th eigenvalues of **M.** (10) is an N-dimensional ellipsoid equation, with the length and the direction of its N-th semiaxis given by $a\sqrt{\lambda_i}$ and $\mathbf{v_i}$, respectively. The projection of the cross section to the N dimensional subspace is a hyper ellipsoid also centered at $\mathbf{x_c}$. By choosing different values of a, denoted here as a_i, we can obtain a group of similar hyper ellipsoids which have the lengths and directions of the semiaxes given by $a_i\sqrt{\lambda_i}$ and $\mathbf{v_i}$, respectively. As indicated in Fig.1, the smaller the value of a^2 is, the smaller will be the area of the projected hyper ellipsoid. The principle of GD is to search the symbol vector lying on the smallest hyper ellipsoid, i.e. having the minimum value of $f(\mathbf{x})$. However, finding the smallest hyper ellipsoid containing the symbol vector is not an easy task. If we use the largest hyper ellipsoid which contain all the symbol vectors, then the complexity will be the same as ML decoding. Thus, here we propose to use a smaller hyper ellipsoid and then identify all the symbol vectors inside. The Euclidean distances between these symbol vectors and the received signal vectors are computed. The symbol vector with the smallest Euclidean distance is detected as the transmitted symbol vector. If the ML solution of (3) is inside this hyper ellipsoid, then the GD will have the same solution as the ML decoder.

We propose a fast GD algorithm which is implemented as follows. We use the zero-forcing solution to obtain a hyper ellipsoid, given by

$$f(\mathbf{x}) = a_{zf}^{\ 2} \tag{11}$$

which is on the subspace spanned by the symbol vectors in Ω. We restricted the search among the vector symbols inside this hyper ellipsoid. Since zero forcing is not optimum, the ML solution (i.e., the optimum solution in the sense that the BLER is minimized) should be inside this hyper ellipsoid.

We need to identify all the lattice points (symbol vectors) located inside this hyper ellipsoid. Since the axes of the hyper ellipsoid are not in parallel with those of the lattice points, it is difficult to directly use the surface equation of the hyper ellipsoid to do this. To get around this problem, we propose to use a circumscribed hyper rectangle. We first build up a new N-dimensional rectangular coordinate X' along the semiaxes of the hyper ellipsoid, and use the center point of the hyper ellipsoid as the origin of the coordinate system. Apparently, the apexes of this hyper rectangle can be denoted in the new coordinate as $\mathbf{k_p}' = [x'_{p1}, x'_{p2}, x'_{p3}, \cdots, x'_{pN}]$, where $p = 1, 2, \cdots, 2^N$. Since the length of semiaxis of the hyper ellipsoid in the i-th dimension is $a_{zf}\sqrt{\lambda_i}$, the value of the i-th component x'_{pi} should be either $a_{zf}\sqrt{\lambda_i}$ or $-a_{zf}\sqrt{\lambda_i}$.

By using coordinate transformation algorithm, the coordinates of the apexes in the original lattice point space can be obtained as

$$\mathbf{k_p}^{\mathrm{T}} = \mathbf{V} \cdot \left(\mathbf{k_p}'\right)^{\mathrm{T}} + \mathbf{x_c} \tag{12}$$

where $\mathbf{V}$ is the eigenvector matrix given by (8). Here, $\mathbf{V}$ serves as the transformation matrix [6], transforming the coordinates of the N-dimensional rectangular coordinate X' to the coordinates of the original lattice space rectangular coordinate X.

It can be shown that the upper bound and lower bound of the j-th component of all the apexes are given by

$$\begin{aligned} x_{i\_\max} &= x_{ci} + \sum_{q=1}^{N} \left| v_{qi} a_{zf} \sqrt{\lambda_q} \right| \\ x_{i\_\min} &= x_{ci} - \sum_{q=1}^{N} \left| v_{qi} a_{zf} \sqrt{\lambda_q} \right| \end{aligned} \tag{13}$$

Because the circumscribed hyper rectangle encloses the hyper ellipsoid, the lattice point located inside the hyper ellipsoid must be within the boundaries given by (13).

Since the axes of the circumscribed hyper rectangle are still not in parallel with the axes of the lattice, it is not easy to use it to find the lattice points inside. So based on the upper and lower bounds given by (13), we make another hyper rectangle which has its axes in parallel with those of the lattice to enclose the old hyper rectangle. This hyper rectangle may be much larger than the original hyper ellipsoid and, in some conditions, the number of the lattice points inside it may be very large.

The axes of this new hyper rectangle are in parallel with the axes of the lattice, so we can easily use the opposite sides of the rectangle to find the coordinates of the lattice points between them. We do this for all opposite sides of the hyper rectangle and the combinations of these coordinates obtained identify the lattice points inside. Since the hyper rectangle may be much larger than the hyper ellipsoid, in order to find out the exact lattice points inside the hyper ellipsoid, we need to use the equation of the hyper ellipsoid (11) as follows. We substitute all the coordinates, except the coordinates for the axis on which most points are lying within the boundaries, into the equation of the hyper ellipsoid (11) to find the actual coordinates of the points inside the hyper ellipsoid. The whole set of complete coordinates identify the lattice points inside the hyper ellipsoid therefore can be found. The value of a^2 for each symbol vector found is then computed and the one with minimum a^2 is decoded as the transmitted symbol vector.

4. Simulation Results and Discussions

Monte Carlo computer simulation has been used to assess the BLER performances and the complexity reduction of the proposed GD algorithm used in the 2×2, 3×3 and 4×4 MIMO systems transmitting 64QAM over a block-fading channel. Complexity is measured by the number of symbol vectors used in the decoding process. In ML decoding, all symbol vectors in Ω are used for computing the Euclidean distances, so the implementation complexity of the ML decoder increases exponentially with the numbers of transmit and receiver antennas and also the level of modulation used in the MIMO system. With our proposed GD algorithm, the symbol vectors used for computing the Euclidean distances are limited to those inside the hyper ellipsoid determined by the zero forcing solution. The results on complexity reductions at a BLER of about 10^{-2} for the MIMO systems studied are shown in Table 1. It can be seen that, for the 2×2, 3×3 and 4×4 MIMO systems transmitting 64QAM, the respective complexity reductions are 97.5%, 98.4% and 99.1%, compared with those of using an ML decoding algorithm. Thus a significant reduction in complexity can be obtained by using our proposed GD implementation.

Table 1 Complexity reduction of proposed GD decoding compared to ML decoding (indicated as percentage)

Modulation	2×2	3×3	4×4
64QAM	97.5%	98.4%	99.1%

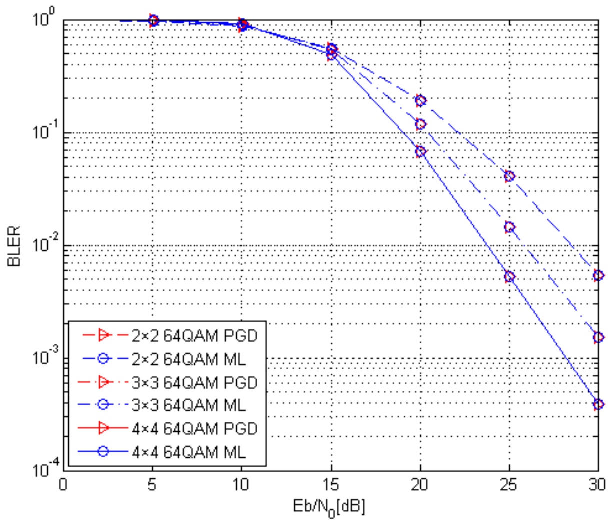

Fig.2 BLER performances of different MIMO systems transmitting 64QAM using proposed GD and ML decoding algorithms

The BLER performances of the 2×2, 3×3 and 4×4 MIMO systems transmitting 64QAM using the proposed and ML decoding algorithms are shown in Fig.2. It can be seen that, at all the SNR values used for tests, the proposed GD and ML decoding algorithms have about the same BLER performances for all the MIMO systems studied, thus our proposed GD algorithm is optimum in the sense that it minimize the BLER. This is expected because both algorithms attempt to find the same symbol vector with the minimum distance and decode it as the transmitted symbol vector. However, our proposed GD can speed up the decoding process by narrowing the search region.

5. Conclusions

The paper has introduced a fast GD algorithm for MIMO systems. It uses the channel matrix and the zero forcing solution to obtain a hyper ellipsoid which is used a search region for the GD solution. Computer simulation results have indicated that our proposed algorithm has exactly the same error performance as an ML decoding algorithm, but it can achieve complexity reductions of more than 97% over the ML decoder.

References

[1] G. J. Foschini, M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antenna," *Wireless Personal communications*, no.6, pp.315-335,1998.

[2] A. Goldsmith, S. A. Jafar, N. Jindal and S. Vishwanath, "Capacity limits of MIMO channels," *IEEE Journal on Selected areas in Communications*, vol. 21, no. 5, pp. 684–702, Jun. 2003.

[3] D. Seethaler, H. Artes, and F. Hlawatsch, "Efficient near-ml detection for mimo channels: The sphere-projection algorithm," in Proc. IEEE Global Telecommunications Conf., vol. 4, pp. 2089–2093, Dec. 2003.

[4] M. Samuel and M. P. Fitz, "Geometric Decoding Of PAM and QAM Lattices," in *Proc. IEEE Global Telecommunications Conf.*, pp. 4247–4252, Nov.2007.

[5] A. Paulraj, R. Nabar and D. Gore, "Introduction to Space-Time Wireless Communications," Cambridge University Press, May 2003.

[6] E. Kreyszig, "Advanced engineering mathematics," John Wiley and sons, Inc, 1962.